|  |  |  |  |
| --- | --- | --- | --- |
|  | | logo_black.gif | |
|  | 200924881 |  | 1,362 |
|  | GEOG5032/GEOG5042: GIS Programming Core Skills |  |  |
|  | Assessment 2 |
|  |  |

|  |
| --- |
|  |
| 1. |
| 2. |
| 3. |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

**Programming Assessment 2**

Code & Data:

This zip file, contains the code which is split into two .py files titled ‘DRUNKCHECKMODEL’ and ‘DRUNKCHECKFRAMEWORK’. The ‘DRUNKCHECKMODEL’ .py file holds the main body of the code but the ‘DRUNKCHECKFRAMEWORK’ holds the functions that present movement within the animation and is essential for the code to work.

The only data inputted into this code was the .txt file available on the School of Geography programming website for this project. This is also included within the zip file, ensure this is saved in the same area as the .py files always to ensure the code runs.

All code and .txt files are also available from my personal GitHub: <https://github.com/JessicaSwain/drunks_going_home>

Code Instructions:

The code has been commented thoroughly so the flow is easier to understand.

To run the code simply press the run button on the toolbar in spyder.

This will allow the pop up window.

*(If no pop up window appears, ensure within preferences in python the IPython console graphics backend is set to automatic instead of inline.)*

In the pop up window, click ‘run model’ on the drop-down menu.

This will then show the drunks moving within the environment, if any stop moving this means they have reached home and their journey has ended.

Close the pop-up window at any time to stop the model from running.

By running the model, a .txt file is set up and results are written within here.

UML Diagram:



Figure 1: UML of the drunk model

Documentation:

1. **Context**

The simple context of this model is to present an environment of a pub and 25 surrounding homes. This is populated by 25 individuals known within the code as drunks due to their time spent at the pub. Each of these drunken individuals are assigned a house number (e.g. drunk 1 lives at house number 10), and after leaving their pub of choice, they need to navigate themselves home. All 25 drunks are assigned randomly to one of the 4 pubs. Each move is random throughout the environment, but based on their X and Y coordinates not matching their home address. Given this lack of similarity, the drunks are told to move around the environment until they arrive home. Once they finally reach their house they stop moving.

1. **Software Design Thought Process and Development Process**

Thoughts towards the beginning of the design was to keep the model as simple as possible to ensure effective running. Agent based modelling and object-orientated programming do not need to be over complicated to present realism; and this was the focal thought throughout the development of the software. Although it should be noted that representing some reality appeared more difficult than first anticipated.

The development process was mainly a trial and error process. For each section of the expected, a basic section of code would be written. Once the whole model presented basic movement randomly for 25 individuals, the code was then looked at further to include house numbers which matched individuals, this would then impact their movements and where and when they would stop. For each element of code, different conditions and code was attempted; if problems arose these could be analysed and rectified. By gradually making changes, this meant steps could be retraced to prevent broken code being untraceable. Although this design process is much longer than required, this seemed to be the safest way to develop the software.

1. **Issues & Overcoming Issues**

Several issues were faced throughout this project. Some which prevented running of the model. This section will briefly explain the issues and how these were overcome if this was possible. The issue that arose firstly with the software development, was the alignment of the environment -as the image below shows.



Figure 2: Drunk model environment taken from the text file on the website

The issue with this environment is that these do not align accurately. Therefore, developing a .csv file or a systematic way of assigning x and y values to individual homes was difficult due to the lack of symmetry in the environment. To overcome this, I developed a dictionary for both the x values and the y values.

HouseY = {"10": 25, "20": 60, "30": 110, "40": 140, "50": 180…

Figure 3: An example of the dictionary code for the environment

For each house number, an x value was assigned and the same was done for a y value. These were then used when the drunk individuals were commanded to move around the environment randomly. Although this is not a major issue as the model can still function without symmetry, this is an area that could be developed further at a later date.

Linking to the decisions to use x and y coordinates for the homes within the environment is another issue, which focused on the difficulty in the ability to read a text file. To be able to use the numbers within this file to get the agents to move as well as present the environment appeared difficult to succeed with. After a variety of attempts, it was clear from online python sources that using .txt files in a more complicated way and for two uses appeared to be difficult to do correctly. Therefore, the text file was inputted for the purposes of environmental display in the model only. Instead, the interaction amongst the homes, pub and drunks was developed through for loops and lists separately -almost as a separate layer to the environment. This made the model more simplistic, and perhaps more long-winded, but this allowed the model to successfully present movement and animation.

Due to the use of for and if loops and the use of these within each other, the movement was often inaccurate, and code became more complicated to work with. This was difficult to rectify as it would be unclear which for loop was not being used when the model ran. By printing a comment after each loop, it was much easier to rectify any for and if loops that did not work as they should.

For and if loops are just one of the methods used within this experimental process. Dictionaries and other conditional statements such as ‘while’ were also used to successfully present movement. It should be noted, that this is not the final version of this code. The code could be improved, extended and developed further if time and experience had allowed. This code is in its infancy, and with time this code could be used to present differing demands in the situation of interest.

Other issues were not so easy to overcome, for example, ensuring the model animation stayed within the frame was difficult to enforce. Often, the movement from pub to homes meant the movement shifted off the grid, and this was difficult to understand why. Perhaps this could be an improvement if the model was to be improved in the future.

Within the code approach there have been some limitations to fulfilling the demands set. For one, due to the use of dictionaries, lists and strings being able to print these to a text file became complicated. In the code, there is evidence of the need to change the data types to strings for the printing to a text file to be successful. As can be seen in the code, the approach to printing to a text file is correct, but perhaps due to complications within the code, the writing demands causes an issue with the code which then breaks the code.

The code presented in the zip file, is the most current and working animated model. The environment is displayed; the drunks are moving randomly and are attempting to get home. From trialling the model, it appears it takes an incredibly long time for these drunks to find their homes, but when this occurs this will print a comment in the console, for the user to see.

1. **General Sources**

The main sources used includes Stack Overflow, Python for Beginners, and the main Python help page. These were mainly to aid any errors through the trial and error process. Most of the issues I faced people had already solved through these Q&A pages.